REMARKS

Remarks Regarding Claim Objections

The Office Action objects to claim 8 because of the following informalities: 'a computer medium' should recite 'a computer readable medium.' Applicants have accordingly amended claim 8 to recite 'a computer readable medium' instead of 'a computer medium.'

Claim Rejections – 35 U.S.C. § 101

Claims 8-13 stand rejected under 35 U.S.C. § 101 because:

> …the language of the claim raises a questions as to whether the claim is directed merely to an abstract idea that is not tied to a technological art, environment or machine which would result in a practical application producing a concrete, useful, and tangible result to form the basis of statutory subject matter under 35 USC 101.

Applicants respectfully note in response that a claimed invention constitutes patentable subject matter under 35 U.S.C. § 101 if the claimed invention as a whole produces a "useful, concrete and tangible result." *State Street Bank & Trust Co. v. Signature Financial Group Inc.*, 149 F. 3d 1368, 1373, 47 USPQ2d 1596, 1601-02 (Fed. Cir. 1998). Claim 8 produces a result of booting a data processing system. As will be shown below in more detail, booting a data processing system as recited in the preamble of claim 8, is a "useful, concrete and tangible result" under 35 U.S.C. § 101. Claims 8-13 are therefore patentable and the rejections of claims 8-13 under 35 U.S.C. § 101 should be withdrawn.

To be useful, a result must be specific, substantial and credible. M.P.E.P. § 2106(IV)(C)(2)(a) and *In re Fisher*, 421 F.3d 1365, 76 USPQ2d 1225 (Fed. Cir. 2005). A person having ordinary skill in the art will immediately realize that the result of booting a data processing system as claimed in the present application is specific, substantial, and credible. Booting a data processing system allows a user in many cases

to access the data processing system's capabilities. Without booting a data processing system, some or all capabilities will not be accessible. Booting a data processing system is therefore specific to a data processing system, substantial to the operation of a data processing and credible in that data processing systems must be booted for a data processing system's capabilities to be accessed. The result of booting a data processing system as claimed in the present application is therefore useful.

To be concrete the process must have a result that can be substantially repeatable. M.P.E.P. § 2106 (IV)(C)(2)(2)(c). A person having ordinary skill in the art will immediately recognize that the result of booting a data processing system as claimed in the present application is substantially repeatable. Not only is it immediately apparent to one having ordinary skill in the art that booting a data processing system as claimed in the present application is substantially repeatable, but Applicants' present application includes 15 pages and 4 drawings describing various exemplary embodiments that result in booting a data processing system as recited in claims 8-13. Because the result of booting a data processing system as claimed in the present application is substantially repeatable, the result of booting a data processing system is therefore concrete.

To be tangible a result must not be abstract. See M.P.E.P. § 2106(IV)(C)(2)(2)(b). A person having ordinary skill in the art will immediately recognize that booting a data processing system is not abstract. In fact, a data processing system is booted electronically and as such cannot be abstract. The result of booting a data processing system as claimed in the present application is therefore tangible. Because the result of booting a data processing system as claimed in the present application is useful, concrete, and tangible, claim 8 is patentable under 35 U.S.C. § 101. Applicants respectfully request reconsideration of claims 8-13.

Claim Rejections – 35 U.S.C. § 103 Over Coker In View Of Zimmerman

Claims 1, 3, 6, 8, 10, 13-15, 17, and 20 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Coker, *et al.* (U.S. Publication No. 2003/0149837) (hereafter 'Coker')

in view of Zimmerman, *et al.* (U.S. Publication No. 2003/0200290) (hereafter "Zimmerman"). Applicants respectfully traverse each rejection. To establish a prima facie case of obviousness under 35 U.S.C. § 103 the proposed combination of the references must teach or suggest all of Applicants' claim limitations. *In re Royka*, 490 F.2d 981, 985, 180 USPQ 580, 583 (CCPA 1974). As will be shown below in more detail, the proposed combination of Coker and Zimmerman cannot establish a prima facie case of obviousness because the proposed combination does not teach each and every element of the claims of the present application. Claims 1, 3, 6, 8, 10, 13-15, 17, and 20 are therefore patentable and should be allowed. Applicants respectfully request reconsideration of claims 1, 3, 6, 8, 10, 13-15, 17, and 20.

Coker Neither Discloses Nor Suggests Comparing A Current Sequence Of Disk Requests To Data Indicative Of A Previous Sequence Of Disk Requests Or Detecting A Match Between The Current Sequence And The Previous Sequence

The Office Action takes the position that Coker at paragraph 2, and figure 4, discloses the first element of claim 1 and a portion of the second element of claim 1: comparing a current sequence of disk requests to data indicative of a previous sequence of disk requests and detecting a match between the current sequence and the previous sequence. Applicants respectfully note in response, however, that what Coker at paragraph 2, in fact discloses is:

> [0002] The claimed invention relates generally to the field of digital data storage systems, and more particularly but not by way of limitation, to an apparatus and method for optimizing the transfer of data between a host device and a data storage device through dynamic detection of access patterns in the blocks of data requested by the host device.

And Coker describes figure 4 at paragraphs 42-46 as follows:

> [0042] FIG. 4 provides a flow chart for a DATA TRANSFER routine 200, illustrative of steps carried out by the interface circuit 124 in accordance with preferred embodiments of the present invention. An initialization step takes place at 202 during which the disc drive 100 is configured for use

and various parameters are initially selected. Thereafter, the disc drive 100 proceeds with normal operation during which access commands are issued by the host device and received by the interface circuit 124, as indicated at step 204.

[0043] As will be recognized, during operation the host device issues access commands from time to time at a rate and in an order as determined by the needs of the host. Thus, there can be times when access commands are issued relatively infrequently and other times when a large number of access commands are issued in rapid succession. Each read access command identifies the LBAs on the discs 108 the contents of which the host requires the disc drive 100 to retrieve. Each write access command includes the write data that the host requires the disc drive 1100 to store and identifies the specific LBAs in which the disc drive is to store the write data. The interface circuit 124 creates a command node as a sortable access instruction in the command queue 131 for each new command.

[0044] Decision step 206 next determines whether the new access command is a read command. If so, the routine executes a MODE DETECT subroutine 208 as discussed below. On the other hand, if the most recently received access command is a write command, the flow passes to step 210 where the drive performs write command processing, as desired. For example, if write caching is employed the interface circuit 124 will report a command complete status to the host device, temporarily store the write data in the buffer 130, and schedule the writing of the data in the near future. If write caching is not employed, then the interface circuit 124 will give priority to the write command and schedule the writing of the data before continuing with other steps in the flow.

[0045] Continuing with the routine of FIG. 4, the interface circuit 124 performs a command sort strategy at step 212 to identify the next appropriate command node to execute in the command queue 131, and proceeds to execute this command at step 214. It will be noted that the receipt of new access commands and the rate at which commands are executed are asynchronous; thus, FIG. 4 has been provided with return loops to show continued operation of the above steps until the disc drive 100 is turned off or enters an idle mode.

[0046] The MODE DETECT subroutine of FIG. 5 generally operates to evaluate current data access patterns of the host device and dynamically configure the disc drive 100 accordingly to optimize data transfer performance. At step 216 the most recently received (latest) read command is compared to a history of recent read commands. This is preferably carried out using a read command history table as shown in FIG. 6.

That is, Coker at paragraph 2, discloses an apparatus and method for optimizing the transfer of data between a host device and data storage device and Coker at figure 4, discloses a subroutine that operates to evaluate current data access patterns and dynamically configure the disc drive accordingly to optimize data transfer performance. Neither Coker's apparatus and method for optimizing the transfer of data between a host device and data storage device nor Coker's subroutine that operates to evaluate current data access patterns and dynamically configure the disc drive accordingly to optimize data transfer performance discloses comparing a current sequence of disk requests to data indicative of a previous sequence of disk requests and detecting a match between the current sequence and the previous sequence as claimed in the present application. Coker's 'subroutine' is described at paragraph 67-68 as a subroutine for updating a history table with new data associated with a read command when the read command overlaps a data range block associated with a previous read command. Detecting a match between the current sequence of disk requests and the previous sequence of disk requests as claimed in the present application however, occurs so that a copy of data blocks accessed during the current sequence can be stored in a contiguous portion of the disk. Here there is a subtle, yet crucial distinction: updating a history table does not disclose storing a copy of data blocks in a contiguous portion of the disk as claimed in the present application. In fact, Coker does not disclose, at these reference points or anywhere else, a 'contiguous portion' as claimed in the present application. It cannot be said then that Coker's apparatus and method for optimizing the transfer of data between a host device and data storage device nor Coker's subroutine that operates to evaluate current data access patterns and dynamically configure the disc drive accordingly to optimize data transfer performance discloses or suggests comparing a current sequence of disk requests to data indicative of a previous sequence of disk requests or detecting a match between the current sequence and the previous sequence as claimed in the present application. The Office Action therefore cannot establish a prima facie case of obviousness. The rejections of claims 1, 3, 6, 8, 10, 13-15, 17, and 20 should be withdrawn, and the claims should be allowed.

Zimmerman Neither Discloses Nor Suggests Storing A Copy Of Data Blocks Accessed During The Current Sequence In A Contiguous Portion Of The Disk Or Responsive To A Subsequent Request For Data In The Disk Sequence, Mapping The Request To The Sequential Portion Of The Disk And Servicing The Request From Data In The Sequential Portion

The Office Action takes the position that Zimmerman at paragraph 42, and figure 4, discloses the following portion of the second element of claim 1 and the third element of claim 1: storing a copy of data blocks accessed during the current sequence in a contiguous portion of the disk and responsive to a subsequent request for data in the disk sequence, mapping the request to the sequential portion of the disk and servicing the request from data in the sequential portion. Applicants respectfully note in response, however, that what Zimmerman at paragraph 42, in fact discloses is:

> [0042] The learning process 450, is executed if the streaming module 26 cannot find the sector sequence file 22. In one embodiment, the sector sequence file is comprised of a list of sectors that the O/S must read in order to complete the transmission of the disk image transmission. In another embodiment, the sector sequence file is comprised not only the list of sectors but also the sequentially stored actual data contained in the listed sectors. In yet another embodiment, the sector sequence file is comprised a single file including a plurality sector lists and corresponding sector data, such that the simultaneous streaming of different data sets to different sets of requesting clients may be supported. A benefit of storing the actual data is that reading a sequential file is much faster than random reads, and it will in turn increase the server's drive throughput and the ability to more efficiently support multiple streams to multiple clients. One result is that the learned actual data can be read exclusively from the sector sequence file until all of the learned data has been transferred to the client. After that point, the server can revert back to using the virtual drive image. In step 426, the streaming module 26 selects one client from the set of registered clients 2'. In step 428, the selected client is permitted to make its disk access requests conventionally, while the streaming module records in a new sector sequence file all sector access requests the selected client makes to fulfill its desired data download, and, in certain embodiments, the requested data itself. In step 430, the selected client informs the streaming module 26 that it has completed its download. At this point, the new sector sequence file is stored on the virtual drive 8, and the streaming process is resumed at step 408. Described below is a use of the streaming process in network booting applications. If the learning

> process 450 is required to create a sector sequence file 22 in the context of network booting, the selected client is permitted to boot conventionally, using the inventive MBR 24 and drivers (30 and 32), while the streaming module 26 records in a new sector sequence file all sector access requests the selected client makes while booting, and, in some embodiments, the actual data requested.

That is, Zimmerman at paragraph 42 and figure 4, discloses recording in a new sector sequence file sector access requests and the requested data. Zimmerman's recording in a new sector sequence file sector access requests and the requested data does not disclose or suggest storing a copy of data blocks accessed during the current sequence in a contiguous portion of the disk and responsive to a subsequent request for data in the disk sequence, mapping the request to the sequential portion of the disk and servicing the request from data in the sequential portion as claimed in the present application. Storing a copy of data blocks accessed during the current sequence as claimed in the present application occurs in response to detecting a match between the current sequence and the previous sequence of disk requests. At no point does Zimmerman disclose or suggest detecting a match between a current sequence and previous sequence of disk requests as claimed in the present application. Because Zimmerman does not disclose or suggest detecting a match between the current sequence and the previous sequence of disk requests as claimed in the present application Zimmerman cannot disclose or suggest storing a copy of data blocks in response to detecting such a match. It cannot be said therefore that Zimmerman's recording in a new sector sequence file sector access requests and the requested data discloses or suggests storing a copy of data blocks accessed during the current sequence in a contiguous portion of the disk and responsive to a subsequent request for data in the disk sequence, mapping the request to the sequential portion of the disk and servicing the request from data in the sequential portion as claimed in the present application. The Office Action therefore cannot establish a prima facie case of obviousness. The rejections of claims 1, 3, 6, 8, 10, 13-15, 17, and 20 should be withdrawn, and the claims should be allowed.

Claim Rejections – 35 U.S.C. § 103 Over The Combination Of Coker And Zimmerman In View of Hung

Claims 2, 9, and 16 stand rejected for obviousness under 35 U.S.C. § 103(a) as being unpatentable over the combination of Coker and Zimmerman in view of Hung (U.S. Patent No. 5,247,653) (hereafter 'Hung'). To establish a prima facie case of obviousness, the proposed combination of the references must teach or suggest all of the claim limitations of dependent claims 2, 9, and 16. *In re Royka*, 490 F.2d 981, 985, 180 USPQ 580, 583 (CCPA 1974). Dependent claims 2, 9, and 16 depend from independent claims 1, 8, and 14 and include all the limitations of the independent claims from which they depend. In rejecting dependent claims 2, 9, and 16, the Office Action relies on the combination of Coker and Zimmerman as disclosing each and every element of independent claims 1, 8, and 14. As shown above, the combination of Coker and Zimmerman in fact does not disclose each and every element of independent claims 1, 8, and 14. Because the combination of Coker and Zimmerman does not disclose each and every element of independent claims 1, 8, and 14, the combination of the combination of Coker, Zimmerman, and Hung cannot possibly disclose each and every element of dependent claims 2, 9, and 16. The proposed combination Coker, Zimmerman, and Hung, therefore, cannot establish a prima facie case of obviousness, and the rejections 35 U.S.C. § 103(a) should be withdrawn.

Claim Rejections – 35 U.S.C. § 103 Over The Combination Of Coker And Zimmerman In View of Lee

Claims 4-5, 12, and 19 stand rejected for obviousness under 35 U.S.C. § 103(a) as being unpatentable over the combination of Coker and Zimmerman in view of Lee (U.S. Publication No. 2004/0260909) (hereafter 'Lee'). To establish a prima facie case of obviousness, the proposed combination of the references must teach or suggest all of the claim limitations of dependent claims 4-5, 12, and 19. *In re Royka*, 490 F.2d 981, 985, 180 USPQ 580, 583 (CCPA 1974). Dependent claims 4-5, 12, and 19 depend from independent claims 1, 8, and 14 and include all the limitations of the independent claims from which they depend. In rejecting dependent claims 4-5, 12, and 19, the Office

Action relies on the combination of Coker and Zimmerman as disclosing each and every element of independent claims 1, 8, and 14. As shown above, the combination of Coker and Zimmerman in fact does not disclose each and every element of independent claims 1, 8, and 14. Because the combination of Coker and Zimmerman does not disclose each and every element of independent claims 1, 8, and 14, the combination of the combination of Coker, Zimmerman, and Lee cannot possibly disclose each and every element of dependent claims 4-5, 12, and 19. The proposed combination Coker, Zimmerman, and Lee, therefore, cannot establish a prima facie case of obviousness, and the rejections 35 U.S.C. § 103(a) should be withdrawn.

Claim Rejections – 35 U.S.C. § 103 Over The Combination Of Coker And Zimmerman In View of Brady

Claims 7, 11, and 18 stand rejected for obviousness under 35 U.S.C. § 103(a) as being unpatentable over the combination of Coker and Zimmerman in view of Brady (U.S. Patent No. 5,758,050) (hereafter 'Brady'). To establish a prima facie case of obviousness, the proposed combination of the references must teach or suggest all of the claim limitations of dependent claims 7, 11, and 18. *In re Royka*, 490 F.2d 981, 985, 180 USPQ 580, 583 (CCPA 1974). Dependent claims 7, 11, and 18 depend from independent claims 1, 8, and 14 and include all the limitations of the independent claims from which they depend. In rejecting dependent claims 7, 11, and 18, the Office Action relies on the combination of Coker and Zimmerman as disclosing each and every element of independent claims 1, 8, and 14. As shown above, the combination of Coker and Zimmerman in fact does not disclose each and every element of independent claims 1, 8, and 14. Because the combination of Coker and Zimmerman does not disclose each and every element of independent claims 1, 8, and 14, the combination of the combination of Coker, Zimmerman, and Brady cannot possibly disclose each and every element of dependent claims 7, 11, and 18. The proposed combination Coker, Zimmerman, and Brady, therefore, cannot establish a prima facie case of obviousness, and the rejections 35 U.S.C. § 103(a) should be withdrawn.

Relations Among Claims

Independent claim 1 claims method aspects of loading data from disk in a data processing system according to embodiments of the present invention. Independent claims 8 and 14 respectively claim computer program product and system aspects of loading data from disk in a data processing system according to embodiments of the present invention. Claim 1 is allowable for the reasons set forth above. Claims 8 and 14 are allowable because claim 1 is allowable. The rejections of claims 8 and 14 therefore should be withdrawn, and claims 8 and 14 should be allowed.

Claims 2-7, 9-13, and 15-20 depend respectively from independent claims 1, 8, and 14. Each dependent claim includes all of the limitations of the independent claim from which it depends. Because the combination of Coker and Zimmerman does not disclose or suggest each and every element of the independent claims, so also the combination of Coker and Zimmerman cannot possibly disclose or suggest each and every element of any dependent claim. The rejections of Claims 2-7, 9-13, and 15-20 therefore should be withdrawn, and these claims also should be allowed.

Conclusion

Claims 8-13 stand rejected under 35 U.S.C. § 101 as being unpatentable subject matter. Claims 8-13 produce a concrete, tangible, and useful result and are therefore patentable and should be allowed. Applicants respectfully request reconsideration of claims 8-13.

Claims 1, 3, 6, 8, 10, 13-15, 17, and 20 stand rejected under 35 U.S.C. § 103 as obvious over Coker in view of Zimmerman. The combination of Coker and Zimmerman does not teach or suggest each and every element of Applicants' claims. Claims 1, 3, 6, 8, 10, 13-15, 17, and 20 are therefore patentable and should be allowed. Applicants respectfully request reconsideration of claims 1, 3, 6, 8, 10, 13-15, 17, and 20.

Claims 2, 9, and 16 stand rejected under 35 U.S.C. § 103 as obvious over the combination of Coker and Zimmerman in view of Hung. The combination of Coker, Zimmerman, and Hung does not teach or suggest each and every element of Applicants' claims. Claims 2, 9, and 16 are therefore patentable and should be allowed. Applicants respectfully request reconsideration of claims 2, 9, and 16.

Claims 4-5, 12, and 19 stand rejected under 35 U.S.C. § 103 as obvious over the combination of Coker and Zimmerman in view of Lee. The combination of Coker, Zimmerman, and Lee does not teach or suggest each and every element of Applicants' claims. Claims 4-5, 12, and 19 are therefore patentable and should be allowed. Applicants respectfully request reconsideration of claims 4-5, 12, and 19.

Claims 7, 11, and 18 stand rejected under 35 U.S.C. § 103 as obvious over the combination of Coker and Zimmerman in view of Brady. The combination of Coker, Zimmerman, and Brady does not teach or suggest each and every element of Applicants' claims. Claims 7, 11, and 18 are therefore patentable and should be allowed. Applicants respectfully request reconsideration of claims 7, 11, and 18.

The Commissioner is hereby authorized to charge or credit Deposit Account No. 50-0563 for any fees required or overpaid.

Respectfully submitted,

Date: December 19, 2006 By:

John Biggers
Reg. No. 44,537
Biggers & Ohanian, LLP
P.O. Box 1469
Austin, Texas 78767-1469
Tel. (512) 472-9881
Fax (512) 472-9887
ATTORNEY FOR APPLICANTS